



# Rajouter un utilisateur à un groupe

- Objet : Apprendre comment rajouter un utilisateur à un groupe, comment vérifier que c'est bien fait, et quelques autres subtilités
- Niveau requis :  
[débutant](#)
- Commentaires : Une opération basique, mais qui comporte quelques subtilités à laquelle nous sommes nombreux à nous être frotté...
- Débutant, à savoir :
  - [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
  - [Les droits sous UNIX](#)
  - [La commande usermod](#)
  - [La commande adduser](#)
  - [Le super utilisateur \(root\)](#)
- Suivi :  
[à-tester](#)
  - Création par  [captntfab](#) 13/08/2023
  - Testé par <...> le <...> 
- Commentaires sur le forum : [Lien vers le forum concernant ce tuto](#) <sup>1)</sup>

## Introduction

L'exécution de programme, l'accès à des fichiers ou même à des périphériques nécessite des droits qui peuvent être conditionnés par l'appartenance à un groupe.

## Prérequis

Pour rajouter un utilisateur à un groupe, il avoir les droits de super-utilisateur (cf. à savoir en entête), et il faut que le groupe existe.

Pour vérifier que le groupe existe:

```
getent group le_nom_du_groupe
```

Si cette commande n'affiche rien, alors le groupe n'existe pas. Si cette commande affiche une ligne ressemblant à l'une des lignes suivantes, alors le groupe existe:

```
adm:x:4:  
docker:x:135:captntfab
```

## Lister les membres d'un groupe

La commande ci-dessus permet de lister les membres du groupes, ils sont après le troisième :. On peut obtenir directement la liste des membres via la commande suivante:

```
getent group le_nom_du_groupe | cut -d: -f4
```

Exemple:

```
getent group docker | cut -d: -f4
```

```
captnfab,mattux
```

Si un utilisateur n'est pas dans la liste, il n'est pas membre du groupe, s'il est dans la liste, il est membre du groupe.

## Lister les groupes d'un processus

Lorsqu'un programme est lancé, il hérite des droits, et donc des groupes, du programme parent.

Par exemple, - Si un utilisateur debian se connecte via le gestionnaire de session GDM3,



1. cela a pour effet de lancer son gestionnaire de fenêtre `gnome-shell`,
2. alors le processus `gnome-shell` lancé aura les droits de l'utilisateur `debian`, et des groupes auxquels appartient l'utilisateur `debian`
3. c'est l'étape de `*login*` (connexion) qui permet cela.

- Si ensuite depuis `gnome-shell` l'utilisateur lance `gnome-terminal`,

1. alors le processus créé héritera des mêmes droits,
2. et il lancera un shell, par exemple `bash`, qui à son tour héritera des mêmes droits.

Les groupes d'un shell peuvent être listés facilement en tapant la commande

```
id -nG
```



Il est important de retenir que les droits ont été définis à la connexion, en se basant sur les droits de l'utilisateur, et ne sont plus changés ensuite. En particulier, si les droits d'un utilisateur changent, les droits des processus lui appartenant restent inchangés.

## Rajouter un utilisateur à un groupe

Pour rajouter un utilisateur à un groupe qui existe déjà, on utilise par exemple `usermod`, ou `adduser`:

Exemple, pour rajouter l'utilisateur `captnfab` au groupe `docker`:

```
usermod --append --groups=docker captnfab
```

Que l'on peut abrégé en

```
usermod -aG docker captnfab
```

Ou alternativement

```
adduser captnfab docker
```

## Obtenir un processus avec plus de droits sans se déconnecter

Cas d'exemple, le shell est un processus lancé par `captnfab`, lequel au moment du lancement n'appartenait pas au groupe `docker`, mais maintenant si.

Logiquement, le shell n'appartient pas au groupe `docker`, comme en témoigne la commande `id -nG`.

On peut lancer des programmes `*setuid root*`, qui s'exécutent en `root` quel que soit l'utilisateur qui les lance et peuvent donc obtenir plus de permissions (comme lors du login).

C'est le cas de `newgrp`, `su` ou `sudo`.

NB: dans la suite, `exec` est utilisé pour dire que le processus lancé remplace l'actuel plutôt que de devenir un fils de l'actuel. Cela évite d'avoir des shells imbriqués.

### **newgrp**

Pour lancer un nouveau shell ayant suffisamment de droits, on peut faire

```
exec newgrp docker
```

`newgrp docker` lance un shell avec comme nouveaux droits : droits du processus parent + groupe `docker` (si l'utilisateur appartient au groupe `docker`)

### **su / sudo**

Les droits sont donnés lors du `*login*`, qu'à cela ne tienne, pour les réinitialiser, il suffit de se logger à nouveau via par exemple

```
exec su - $USER
```

ou

```
exec sudo -s -u $USER
```

## Résolution de problème

### Je suis dans le groupe mais j'ai pas les droits

Ce n'est pas seulement l'utilisateur qui doit avoir les droits, mais surtout le processus.

Si des droits ont été ajoutés à un utilisateur pendant qu'il était connecté, les programmes lancés ne vont pas bénéficier de ces droits.

La méthode la plus simple pour résoudre ce problème consiste à redémarrer ou à se déconnecter/reconnecter. Sinon, utiliser `newgrp`, `su`, ou `sudo` (cf. ci-dessus)

<sup>1)</sup>

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:systeme:droit-rajouter-un-utilisateur-a-un-groupe>

Last update: **13/08/2023 14:48**

